

Amendments to the Claims

No claim is amended. A complete listing of the current pending claims is provided below for the Examiner's convenience.

1. (Previously Presented) A method of optimizing a software program for a target processor to meet performance objectives, where the software program is coded in a high-level language, the method comprising the steps of:

(a) optimizing the software program such that a resulting first optimized form of the software program is completely independent of the target processor and is at least partially coded in the high-level language, determining a first performance profile for the first optimized form of the software program, and comparing the first performance profile with the performance objectives;

(b) based on results of comparing the first performance profile with the performance objectives, if the performance objectives are not met by the first optimized form of the software program, then optimizing the first optimized form of the software program such that a resulting second optimized form of the software program includes at least one portion that is dependent on the target processor and is coded in the high-level language, wherein the at least one portion of the second optimized form of the software program is less than an entirety of the second optimized form; and

(c) flagging the at least one portion to indicate that the at least one portion is dependent on the target processor if the first optimized form of the software program is optimized to create the second optimized form of the software program;

wherein the acts of optimizing are performed such that the first and second optimized forms of the software program are progressively more dependent on the target processor.

2. (Previously Presented) The method of claim 1, further comprising steps of:

(b1) determining a second performance profile for the second optimized form of the software program, and comparing the second performance profile with the performance objectives.

3. (Previously Presented) The method of claim 1, further comprising:

(d) optimizing the second optimized form of the software program such that a resulting third optimized form of the software program is at least partially dependent on the target processor and includes portions coded in a low-level language of the target processor.

4. (Previously Presented) The method of claim 1 in which step (a) comprises the act of deriving a floating point implementation.

5. (Previously Presented) The method of claim 1 in which step (a) comprises the act of deriving a fixed point implementation.

6. (Previously Presented) The method of claim 5 in which the act of deriving the fixed point implementation comprises the act of processing qualification.

7. (Previously Presented) The method of claim 5 in which the act of deriving the fixed point implementation comprises the act of implementation sizing.

8. (Previously Presented) The method of claim 1 in which step (a) comprises the act of implementing reference code.

9. (Previously Presented) The method of claim 8 in which the act of implementing reference code comprises code profiling.

10. (Previously Presented) The method of claim 1 in which step (b) comprises the act of optimization predicted to improve resulting assembly code.

11. (Previously Presented) The method of claim 1 in which step (b) comprises the act of tuning low-level functions.

12. (Previously Presented) The method of claim 3 in which step (d) comprises the act of manual assembly optimization.

13. (Previously Presented) The method of claim 1 in which step (b) comprises the act of feature tuning.

14. (Previously Presented) A computer-readable medium comprising a sequence of instructions which, when executed by a processor, causes the processor to execute a method for optimizing a software program for a target processor to meet performance objectives, where the software program is coded in a high-level language, the method comprising the steps of:

(a) optimizing the software program such that a resulting first optimized form of the software program is completely independent of the target processor and is at least partially coded in the high-level language, determining a first performance profile for the first optimized form of the software program, and comparing the first performance profile with the performance objectives;

(b) based on results of comparing the first performance profile with the performance objectives, if the performance objectives are not met by the first optimized form of the software program, then optimizing the first optimized form of the software program such that a resulting second optimized form of the software program includes at least one portion that is dependent on the target processor and is coded in the high-level language, wherein the at least one portion of the second optimized form of the software program is less than an entirety of the second optimized form; and

(c) flagging the at least one portion to indicate that the at least one portion is dependent on the target processor if the first optimized form of the software program is optimized to create the second optimized form of the software program;

wherein the acts of optimizing are performed such that the first and second optimized forms of the software program are progressively more dependent on the target processor.

15. (Previously Presented) The computer-readable medium of claim 14, in which the method further comprises the steps of:

(b1) determining a second performance profile for the second optimized form of the software program, and comparing the second performance profile with the performance objectives.

16. (Previously Presented) The computer-readable medium of claim 14, wherein the method further comprises:

(d) optimizing the second optimized form of the software program such that a resulting third optimized form of the software program is at least partially dependent on the target processor and includes portions coded in a low-level language of the target processor.

17. (Previously Presented) The computer-readable medium of claim 14 in which step (a) comprises the act of deriving a floating point implementation.

18. (Previously Presented) The computer-readable medium of claim 14 in which step (a) comprises the act of deriving a fixed point implementation.

19. (Previously Presented) The computer-readable medium of claim 18 in which the act of deriving the fixed point implementation comprises the act of processing qualification.

20. (Previously Presented) The computer-readable medium of claim 18 in which the act of deriving the fixed point implementation comprises the act of implementation sizing.

21. (Previously Presented) The computer-readable medium of claim 14 in which step (a) comprises the act of implementing reference code.

22. (Previously Presented) The computer-readable medium of claim 21 in which the act of implementing reference code comprises code profiling.

23. (Previously Presented) The computer-readable medium of claim 14 in which step (b) comprises the act of optimization predicted to improve resulting assembly code.

24. (Previously Presented) The computer-readable medium of claim 14 in which step (b) comprises the act of tuning low-level functions.

25. (Previously Presented) The computer-readable medium of claim 16 in which step (d) comprises the act of manual assembly optimization.

26. (Previously Presented) The computer-readable medium of claim 14 in which step (b) comprises the act of feature tuning.

27. (Previously Presented) The method of claim 1, wherein the second optimized form of the software program includes the at least one portion that is dependent on the target processor and another portion that is independent of the target processor.

28. (Previously Presented) The method of claim 1, wherein the act of optimizing the first optimized form of the software program uses a subset of the first optimized form.

29. (Previously Presented) The computer-readable medium of claim 14, wherein the second optimized form of the software program includes the at least one portion that is dependent on the target processor and another portion that is independent of the target processor.

30. (Previously Presented) The computer-readable medium of claim 14, wherein the act of optimizing the first optimized form of the software program uses a subset of the first optimized form.

31. (Previously Presented) A system of optimizing a software program for a target processor to meet performance objectives, where the software program is coded in a high-level language, the system comprising:

a processor configured for:

(a) optimizing the software program such that a resulting first optimized form of the software program is completely independent of the target processor and is at least partially coded in the high-level language, determining a first performance profile for the first optimized form of the software program, and comparing the first performance profile with the performance objectives;

(b) based on results of comparing the first performance profile with the performance objectives, if the performance objectives are not met by the first optimized form of the software program, then optimizing the first optimized form of the software program such that a resulting second optimized form of the software program includes at least one portion that is dependent on the target processor and is coded in the high-level language , wherein the at least one portion of the second optimized form of the software program is less than an entirety of the second optimized form; and

(c) flagging the at least one portion to indicate that the at least one portion is dependent on the target processor if the first optimized form of the software program is optimized to create the second optimized form of the software program;

wherein the acts of optimizing are performed such that the first and second optimized forms of the software program are progressively more dependent on the target processor.